Amendments to Claims

Cancel claims 1-34 and 54-65

35. (Currently amended) A method of discovering one or more patterns in a set of k sequences of symbols, called a k-tuple, where k is greater than or equal to two, within an overall set of w sequences having sequence numbers 1, 2, ..., w , the symbols being members of an alphabet, each sequence of symbols having respective lengths L_1, L_2, ..., L_w, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in a sequence;

b) for each of the w sequences ~~sequence~~, grouping the (symbol, position index) pairs by symbol to form a respective master offset table, thus creating w master offset tables;

c) using the w master offset tables, forming a k-tuple table associated with the k-tuple, the table comprising k columns, one of the k columns being a primary column and the remaining (k-1) columns being suffix columns ,
each column corresponding to one of the k sequences;

i) the ~~first,~~ primary ~~,~~ column comprising the (symbol, position index) pairs of a ~~the first,~~ primary ~~,~~ sequence,

ii) the ~~subsequent~~ (k-1) suffix columns comprising (symbol, difference-in-position value) pairs, where the difference-in-position ~~value~~ values are the position differences between all ~~possible~~ like symbols of each remaining sequence of the tuple and the primary sequence of the tuple,

iii) the rows in the k-tuple table resulting from forming all ~~possible~~ combinations of like symbols from each sequence;

d) creating a sorted k-tuple table by performing a multi-key sort on the k-tuple table, the sort keys being selected respectively from the difference-in-position ~~value~~ values of the last suffix column (k^{th} column) through the difference-in-position value of the first suffix column ~~(2^{nd} column)~~;

e) defining a set of patterns by collecting adjacent rows of the sorted k-tuple table whose suffix columns contain identical ~~sets of~~ difference-in-position values, the relative positions of the symbols in each pattern being determined by the primary column position indices, the set of patterns being common to the k sequences.

36. (Original) The method of claim 35 further comprising:

f) deleting all patterns not satisfying a predetermined criteria.

37. (Original) The method of claim 35 further comprising:

f) deleting all patterns shorter than a first predetermined span and longer than a second predetermined span.

38. (Original) The method of claim 35 further comprising:

f) deleting all patterns having fewer than a predetermined number of symbols.

39. (Original) The method of claim 35, further comprising the step of deleting rows from the k-tuple table which do not have suffix indices identical to any other row of the k-tuple table.

40. (Original) The method of claim 35 further comprising the step of deleting rows from the k-tuple table according to predetermined criteria.

41. (Currently amended) The method of claim 40, wherein rows sharing identical suffix column difference-in-position values are deleted from the k-tuple table if there are fewer than N_s such rows ~~sharing identical suffix column difference in position values~~, where N_s is the minimum number of symbols per pattern.

42. (Currently amended) A method of forming a (k+1)-tuple table, wherein a k-tuple table is combined with a sequence of symbols, comprising the steps of:

a) translating the sequence of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in a the sequence of symbols;

b) grouping the (symbol, position index) pairs by symbol to form a ~~respective~~ master offset table;

c) creating the (k+1)-tuple table of k+1 columns , one of the k+1 columns being a primary column and the remaining k columns being suffix columns , by:

i) forming all combinations of like symbols between the primary column of the k-tuple table and the master offset table,

ii) for each such combination, duplicating the corresponding row of the k-tuple table, and appending a (symbol, difference-in-position value) pair corresponding to the difference between the position index of the master offset table and the position index of the primary column.

43. (Currently amended) The method of claim 42 further comprising the step of:

deleting patterns from a k-tuple table common to the k-tuple table and a (k+1)-tuple table, where the (k+1)-tuple table contains all of the sequences of the k-tuple table with one ~~addition~~ additional sequence, by:

a) deleting the suffix column corresponding to a sequence not shared between the two tuple tables, thereby defining a modified table, and

b) deleting rows from the k-tuple table whose suffix columns contain identical sets of difference- in-position values to a row of the modified table.

44. (Currently amended) A method of discovering one or more patterns in a set of k sequences of symbols, called a k-tuple, comprising the steps of:

a) for a first pair of ~~two~~ sequences of the k-tuple

i) translating each sequence of symbols into a table of ordered (symbol, position index) pairs, where the position index of each (symbol, position index) pair refers to the location of the symbol in the sequence;

ii) for each of the paired ~~two~~ sequences, grouping the (symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

iii) forming a Pattern Map comprising an array having (L1 + L2 –1) rows by:

A) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

B) ~~repeatedly~~ storing each (symbol, position index) pair from the first master offset table in a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

iv) defining a parent pattern by populating an output array with the symbols of each (symbol, position index) pair of a row of the Pattern Map, the symbols being placed at relative locations in the parent pattern indicated by the position index of the (symbol, position index) pair; and

v) repeating step iv) ~~d)~~ for each row of the Pattern Map;

b) storing the discovered patterns as arrays of (symbol, position index) pairs;

c) for each subsequent pair of sequences ~~sequence~~ of the k-tuple, replacing the (symbol, position index) pairs of the first sequence of the pair of sequences by the (symbol, position index) pairs of the stored patterns; and

d) repeating steps (a) through (c) for each subsequent pair of sequences until the k-th sequence ~~level k~~ of the k-tuple is reached.

45. (Currently amended) The method of claim 35, further comprising the step of ~~wherein the method of~~ finding all patterns at all levels of support within ~~from~~ a set of sequences by ~~comprises the steps of~~ :

f) forming a tree of nodes, where each node corresponds to each combination of k sequences, and therefore represents a k-tuple, and wherein each node representing a k-tuple is connected to all nodes representing (k+1)-tuples,

each (k+1)-tuple being formed by adding a unique sequence to the k-tuple, where the sequence being added is later in the ordered list of sequences than the latest sequence of the k-tuple;

~~a) forming a tree of nodes, where each node corresponds to each possible combination of sequences in an ordered set of sequences, and also therefore to a corresponding k-tuple ;~~

~~b) organizing the nodes into a tree structure , wherein a node with a k-tuple is connected to all possible nodes containing (k+1) tuples, the (k+1) tuple being formed by adding a unique sequence to the k-tuple, where the sequence being added is later in the ordered list of sequences than the latest sequence of the k-tuple ;~~

g) ~~e)~~ traversing the tree, and at each node visited during traversal, defining a set of patterns by collecting adjacent rows of the sorted k-tuple table whose suffix columns contain identical sets of difference-in-position values, the relative positions of the symbols in each pattern being determined by the primary column position indices, the set of patterns being common to the k sequences.

46. (Original) The method of claim 45, wherein the traversal of the tree of nodes is accomplished via recursion.

47. (Currently amended) The method of claim 45, further comprising the step of:

h) ~~d)~~ removing duplicate patterns at each level of support.

48. (Currently amended) The method of claim 47, wherein the removal of duplicate patterns at each level of support step h) is accomplished by:

i) for each node corresponding to a (k+1)-tuple, identifying the nodes containing k-tuples whose sequences are subsets of the (k+1)-tuple; thereby defining a set of causally-dependent nodes;

ii) locating said causally-dependent nodes;

iii) removing from each said causally-dependent node the patterns in common with the (k+1)-tuple; and

iv) if the k-tuple table in a causally-dependent node is thereby reduced to zero length, removing the corresponding node and all of its descendents from the tree prior to their traversal.

49. (Currently amended) The method of claim 48, wherein locating causally-dependent nodes in step ii) comprises the steps of:

(A) ~~(a)~~ organizing the nodes at level k in the Tuple-tree into a linked list which is ordered from left to right in accordance with the sequence numbers represented by ~~of~~ each tuple; and

(B) ~~(b)~~ searching said linked list for nodes which are causally-dependent on a particular (k+1)-tuple.

50. (Original) The method of claim 48, wherein the nodes located in step ii) are causally-dependent nodes at level k determined with respect to another node at level k, and are thus causally-dependent on a child of the another node at level k.

51. (Currently amended) The method of claim 47, wherein the removal of duplicate patterns at each level of support step h) comprises the steps of:

i) organizing the nodes at level k in the Tuple-tree into a linked list which is ordered from left to right in accordance with the sequence numbers of each tuple;

ii) for each pattern in the current node at level k, storing a "hit list" array of the sequence numbers ~~indices~~ of the sequences containing the pattern;

iii) for all nodes to the right of the current node whose sequence numbers ~~indices~~ are all in the hit list, searching for a duplicate instance of the pattern, and if found, eliminating it; and

iv) making each node the current node, repeating steps (ii) and (iii), in the order of the node's appearance in the linked list.

52. (Currently amended) The method of claim 51, wherein, in step iii), the nodes consistent with the hit list are found using a hash tree, the hash tree having a root and k levels of nodes, the k-th level of the hash tree having a plurality of leaf nodes, the respective level of nodes of the hash tree corresponding to the respective sequence numbers ~~index~~ of a k-tuple, the leaf nodes identifying the k-tuple whose sequence numbers ~~indices~~ correspond to the path from the root to the leaf node, wherein

searching the nodes for pattern duplicates is performed by repeating steps ii) and iii) for each node in the order of the appearance of that node in the hash tree.

53. (Currently amended) The method of claim 45 wherein the traversing step c) itself comprises the steps of:

i) creating a Virtual Sequence Array of patterns found within the sequences, wherein the patterns are termed P-nodes and the tuple nodes are termed T-nodes,

(ii) finding a P-node list corresponding to the location of each ~~the~~ pattern in the primary sequence of that tree node,

iii) ~~ii)~~ searching the P-node list for a duplicate instance of the pattern,

(A) ~~(a)~~ if no duplicate is found:

(1) ~~(i)~~ adding a pointer to the pattern of ~~to~~ the current T-node pattern array,

(2) ~~(ii)~~ finding all locations of the pattern within the Virtual Sequence Array,

(3) ~~(iii)~~ adding a pointer to the pattern to each corresponding P-node array;

(B) ~~(b)~~ if a duplicate pattern is found:

(1) ~~(i)~~ ignoring the pattern if the duplicate pattern was found at support equal to the current level of support,

(2) ~~(ii)~~ if the duplicate pattern was found at a previous level of support, unlinking the duplicate pattern from its previous T-node (if it exists), and relinking the duplicate pattern to the current T-node,

(3) ~~(iii)~~ repeating steps 1) and 2) ~~i) and ii)~~ until all of the children of a T-node have been created, thus insuring that patterns of ~~on~~ that T-node that are at their ultimate level of support are reported, and

(4) ~~(iv)~~ deleting the T-node.

66. (Currently amended) A computer-readable medium containing a plurality of data structures ~~structure~~ useful in controlling a computer system to discover a set of ~~one or more~~ patterns in k ~~two~~ sequences of symbols within an overall set of w sequences,

a number w of ~~the~~ data structures each ~~structure~~ grouping,

for each value of a difference in position between each occurrence of a symbol in one of the sequences and each occurrence of that same symbol in each ~~the~~ other sequence,

the position (position index) in the first sequence of each symbol therein that appears in each of the other sequences ~~second sequence~~ at that difference-in-position value;

a first additional data structure comprising columns and rows, the columns comprising (symbol, position index) pairs and (symbol, difference-in-position value) pairs; and

a second additional data structure comprising a row-sorted representation of the (symbol, position index) pairs and (symbol, difference-in-position value) pairs contained in the first additional data structure.

67. (Currently amended) The computer-readable medium of claim 66 wherein the second additional data structure further groups, for each value of a difference in position, ~~an indication of~~ the number of symbols in the first sequence that appear in the second sequence at that difference-in-position value.

68. (Newly Added) A computer-readable medium containing instructions for controlling a computer system to discover one or more patterns in a set of k sequences of symbols, called a k-tuple, where k is greater than or equal to two, within an overall set of w sequences having sequence numbers 1, 2, ..., w , the symbols being members of an alphabet, each sequence of symbols having respective lengths L_1, L_2, ..., L_w, by executing a method comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in a sequence;

b) for each of the w sequences, grouping the (symbol, position index) pairs by symbol to form a respective master offset table, thus creating w master offset tables;

c) using the w master offset tables, forming a k-tuple table associated with the k-tuple, the table comprising k columns, one of the k columns being a primary column and the remaining (k-1) columns being suffix columns,
each column corresponding to one of the k sequences;

i) the primary column comprising the (symbol, position index) pairs of a primary sequence,

ii) the (k-1) suffix columns comprising (symbol, difference-in-position value) pairs, where the difference-in-position values are the position differences between all like symbols of each remaining sequence of the tuple and the primary sequence of the tuple,

iii) the rows in the k-tuple table resulting from forming all combinations of like symbols from each sequence;

d) creating a sorted k-tuple table by performing a multi-key sort on the k-tuple table, the sort keys being selected respectively from the difference-in-position values of the last suffix column (k^{th} column) through the difference-in-position value of the first suffix column;

e) defining a set of patterns by collecting adjacent rows of the sorted k-tuple table whose suffix columns contain identical difference-in-position values, the relative positions of the symbols in each pattern being determined by the primary column position indices, the set of patterns being common to the k sequences.